

07 Let's Make a Deal

In the popular television game show of the 70s, [Let's Make a Deal](#), game show host, Monty Hall, would show a contestant 3 doors. Behind one of the doors was a prize and the contestant would have to guess which one it was. Once the contestant made a guess which door led to the prize, Monty Hall would open one of the other doors showing that the prize was not behind it. The contestant then had the opportunity of changing his or her guess or sticking to the original guess. Monty Hall would then open the door that concealed the prize.

The common place belief was that there was no real advantage in changing one's guess. The reasoning went as follows: since one of the doors was eliminated, the prize would be behind one of the two remaining doors, i.e. the probability of it being behind the door the contestant guessed was 0.5. The probability of the prize being behind the other door was also 0.5. So the prize being behind one of the two doors was just as likely as being behind the other door. It did not make sense to switch.

In 1990, a reader of Parade magazine, wrote to Marilyn vos Savant asking her what the best strategy would be to play this game. Her advice was, that no matter what the initial guess was, to switch. You should read [all the comments](#) that other readers sent her including college professors berating her for her suggestion. But subsequent analysis has shown that she was right.

In this programming assignment you are going to simulate the game show and demonstrate that indeed Marilyn vos Savant gave sensible advice. The following describes the steps that you will be taking:

1. You will prompt the user to enter the number of times he or she wants to play this game.
2. You will create a variable to keep track of the number of times the contestant wins by switching.
3. Generate a random number between 1 and 3 (inclusive) that will denote the door that conceals the *prize*.
4. Generate another random number between 1 and 3 (inclusive) that will denote the *guess* the contestant makes.
5. From those two numbers, compute a number that does not conceal the prize nor is it the contestant's guess. This is the door that is opened by Monty Hall and we shall call it *view*.
6. At this point the contestant changes his mind makes a *newGuess* that is not the original *guess* nor is it the *view*.
7. You will compare the value of the *newGuess* with that of *prize*. If they are the same, the contestant won by switching, and you increment the variable that was keeping track of that.
8. You will run this simulation for however many times the user had specified.
9. To obtain the probability for winning if you switch divide the number of times the contestant wins by switching by the total number of games played.
10. To obtain the probability of winning by not switching subtract the above number from one.

A sample session and output will look as follows:

Enter number of times you want to play: 10

Prize	Guess	View	New Guess
2	2	3	1
1	3	2	1
1	1	2	3
3	1	2	3
3	1	2	3
3	1	2	3
2	2	3	1
2	1	3	2
3	2	1	3
2	1	3	2

Probability of winning if you switch = 0.70

Probability of winning if you do not switch = 0.30

Your output must be in the above format. The column headers and numbers must be printed as shown. The probabilities are expressed correct to 2 places of decimal. You can use the *round()* function to do so.